LA-UR-97-3052

Unstructured 3D Grid Toolbox for Modeling and Simulation

Denise George
Los Alamos National Laboratory

Abstract

Computable 3D grids that accurately represent complex multimaterial geometries are essential for both static and time-dependent modeling and simulation. LaGriT, the grid toolbox developed at Los Alamos provides a sophisticated set of initial grid generation, grid maintenance and grid optimization tools. We present example grids that demonstrate the flexibility of the grid generator. Additionally, we present the results of an electrostatic calculation and a grain growth problem that illustrate the grid optimization features and the utility of the grid server architecture.

1: Introduction

The LaGriT toolbox[1] provides a full set of grid generation and grid optimization tools for constructing and maintaining unstructured grids for both static and dynamic applications. The grid generation tools respect intricate geometries, allow optional user control over node placement, and provide hybrid grid capabilities. The grid optimization tools allow adjustment of node density using refinement, simple and adaptive smoothing, and reconnection techniques. The LaGriT toolbox methods as well as the LaGriT data structures are designed as objects in order to be user accessible and extensible. This design feature promotes the separation of the physical based simulation from the grid maintenance chores, but provides full access to grid data structures when required.

2: Grid Generation

Initial grid generation requires two sets of information. The first set is the geometry information: the bounding surfaces and the material regions. The second set is the grid node and element based information. The application geometry is defined as a union of non-overlapping regions such that the union forms a connected convex solid in 3D space. Each region is bounded by surfaces where each surface may change over time. Surfaces may be analytic surfaces such as planes, cylinders and spheres or sheet surfaces which are sets of connected triangles in 3D space with consistent orientation. Because regions are defined in terms of their bounding surfaces and because surfaces may deform over time, regions are also dynamic. If a surface which begins as an analytic surface at time t=0 deforms, it must be represented at later time steps as a sheet surface. LaGriT supports dynamic surface and region definitions. Regions are defined using a Boolean algebra over oriented surfaces, and because of the completeness of the Boolean algebra, arbitrarily complex geometries can be described. Figure 1 shows a geometry all of whose surfaces are planes. Figure 2 shows a geometry in which all material interfaces are sheet surfaces.

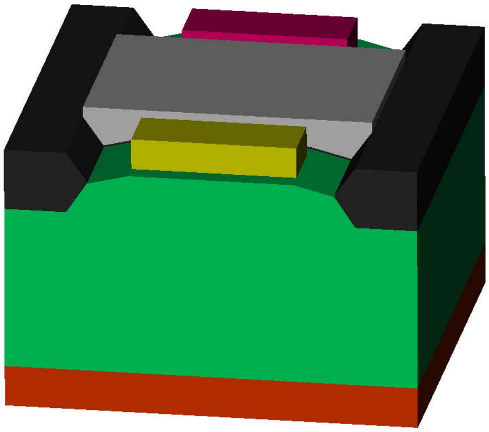

Figure 1. Seven region geometry solid geometry model. The ‘air’ region has been removed for viewing. All bounding surfaces are planes.

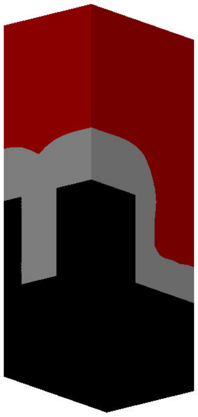

Figure 2. Three region solid geometry model. The two material interfaces are defined by sheet surfaces.

We previously defined grid geometry in terms of combining pre-existing surfaces into material regions. Often a user has no explicit geometry information and must start from a pre-existing grid or a set of nodes. If existing nodes have an associated material, and if interface and external boundary nodes are identified, it is possible to use the LaGriT **extract** capability to extract 2D sheet surfaces that correspond to the material interfaces and external boundaries, and to use these extracted sheet surfaces to recreate the geometry. The geometry for the grain growth model presented in Section 4 was generated by exactly this method.

Once surfaces and regions have been defined, nodes are distributed on surfaces and within regions using a choice of Cartesian, cylindrical, or spherical coordinate systems. Even, proportional, or exact node distribution permits user control over node placement and element aspect ratios. Standard node distribution involves creating a template of points, constructing rays that extend from an origin plane, line, or point to the template of points, and distributing points along these rays within a specified region. LaGriT supports coordinate system redefinition and a set of point rotation, translation, scaling and copy operations. Because of the symmetry of the Figure 1 solid model, Figures 3, 4 and 5 display details of the grid generated from the half model. Figure 3 shows one of the material regions of Figure 1. This material region was constructed from 4 separate geometric subregions. This subdivision allowed for different number of nodes to be distributed along the rays in the different subregions and also allowed for rays to be constructed perpendicular to the region boundaries. The thin subregions have 4 points along each ray and the thicker subregions have 6. The aspect ratio of the elements in the flat portion of the thin area is large and transitions into a more even aspect ratio in the angled portion. Figure 4 is an enlargement that shows the point distribution along rays constructed perpendicular to the interface surfaces on two of the subregions. Figure 5 shows, rotated by 180°, the material region that fits under the region displayed in Figure 3 and displays proportional node placement. Proportional or ratio spacing creates smaller, high aspect ratio elements near the upper surface. The planar surfaces between these regions and the sharp angles of the region boundaries are preserved by the grid algorithm.

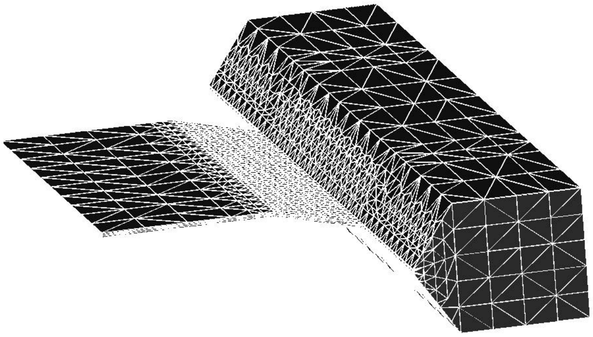

Figure 3. One of the material regions from the first example demonstrating varying aspect ratios.

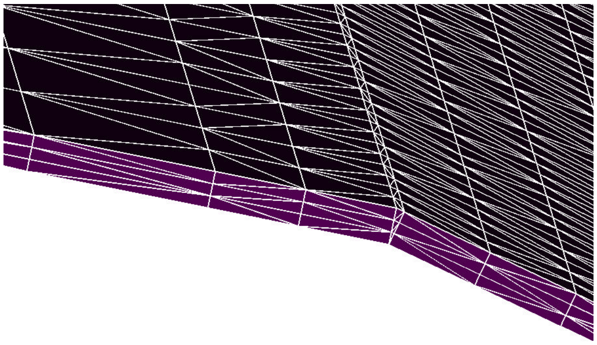

Figure 4. Enlargement of two subregions of the previous region showing perpendicular ray placement and transition from large to smaller aspect ratio elements.

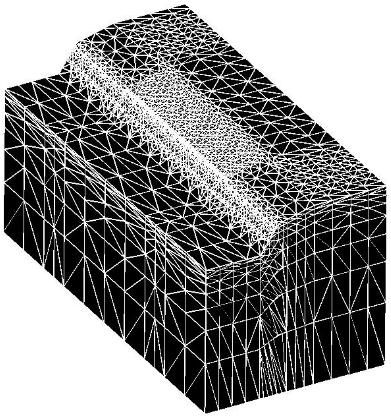

Figure 5. Another of the regions from the solid model displayed in Figure 1. Ratio spacing was used to create smaller elements near the upper material interface and larger elements as we move away from the area of interest. Also visible are the effects of different node template densities.

Sheet surface material interfaces are respected by the grid connection algorithm. Figure 6 is an exploded view of the three materials solid model of Figure 2. An even node distribution has been placed over the entire model by the ray shooting method that also automatically places nodes on material interfaces. In this example, the upper surface of the middle material is an irregular sheet surface. Nodes traced by a previous time step of this surface are also visible about one third the way up the middle material region. These nodes were included in the grid by reading in a 2D grid object and copying the nodes from that object into the 3D grid object. The ability of LaGriT to maintain several concurrent grid objects adds to its flexibility and utility.

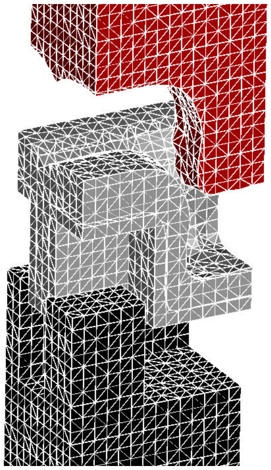

Figure 6. Exploded view of a 3 region geometry constructed using sheet surfaces as material interfaces.

Although LaGriT was designed originally as a Delaunay tetrahedral grid generator, it has been augmented with hybrid grid capabilities. Hexahedral grids span space with fewer nodes and elements than tetrahedral grids. Some applications as best suited by hybrid grids which combine hexahedral grid regions to tetrahedral regions with pyramid elements along the interface.

3: Grid Optimization

Grid optimization combines refinement/de-refinement, smoothing and reconnection tools that modify the grid to provide spatial resolution where needed. Grid refinement adds nodes based on geometric criteria such an edge length, or based on the value of a field variable. For example a grid can be refined where a gradient is steep. The target objects of the refinement can be edges, faces or elements and refinement can be restricted to a subset of the grid. The combination of a large set of selection criteria, the three types of target object, and restriction capabilities provide a very powerful and flexible refinement tool. De-refinement performs the inverse operation by decreasing node density away from areas of interest. Currently, de-refinement is implemented by the **merge** command which merges together a pair of neighboring nodes after verifying that the merge is legal or by the more general **massage** command that specifies a desired edge length and a 'damage' number. We will provide an example of this new command in the next section.

Grid smoothing moves existing nodes without changing grid connectivity. Simple smoothing mode improves the element aspect ratios and distribution of nodes in the mesh. Three smoothing algorithms are available[2]:

1. **esug** Elliptic Smoothing for Unstructured Grids for triangular 2D mesh objects.
2. **mega** Minimum Error Gradient Adaption. This option creates a smoothed grid which is adapted to the standard function $f(x,y,z)=x^2+y^2+z^2$.
3. **geometry** Aspect ratio adaption.

Adaptive smoothing moves the nodes to best capture the behavior of a specified field or to an adaption function supplied by the user. There are two adaptive smoothing algorithms available[3]:

1. **esug** Elliptic Smoothing for Unstructured Grids for triangular 2D meshes. Adaption will be to a specified field or to a user supplied adaption function.
2. **mega** Minimum Error Gradient Adaption. For adaptive smoothing purposes, **mega** is used in conjunction with a user-supplied adaption function or a specified field.

Grid elements may become distorted as a result of refinement and/or smoothing. The reconnection tool interchanges connections to restore the Delaunay criterion.

Although any of the optimization tools may be used independently, combining tools provides powerful optimization capabilities. Figure 7 displays the results of refinement based on a concentration field followed by 2D elliptic smoothing and reconnection. The 2D grid shown was extracted from a 3D grid using the extract grid tool.

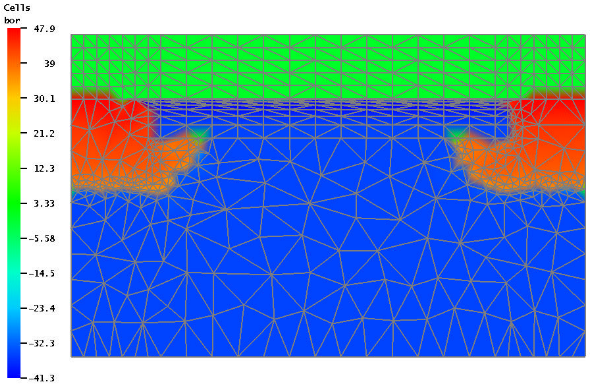

Figure 7. Results of refinement, smoothing and reconnection.

To illustrate the benefit of adaptive grids, we show the effects of solution adaptive grid generation for the calculation of electric fields in non-trivial geometry. Figure 8 shows the interior of a box with a conical intrusion. Displayed at the tip of the cone, where the electric field becomes arbitrarily large, is an isosurface for the component of the electric field aligned with the axis of the cone. As can be seen, this isosurface is nonsmooth due to lack of resolution in this area of highest solution error. For clarity, only surface triangles are displayed; however, the calculation was done on volume tetrahedra. In Figure 9 we show the same view and electric field component isosurface for the solution on a grid that has been adapted to an *a posteriori* error estimate based on estimated second derivatives of the solution. The isosurface is smoother, indicating that adaptive error-based smoothing has successfully attracted the solution elements to the critical area near the tip of the cone. Although we do not display the maximum resolvable electrical field component, it is approximately ten times higher than on the unadapted grid. As a finite solver was used to solve Laplace's equation, the reconnection tool was called to restore the required Delaunay condition on the grid.

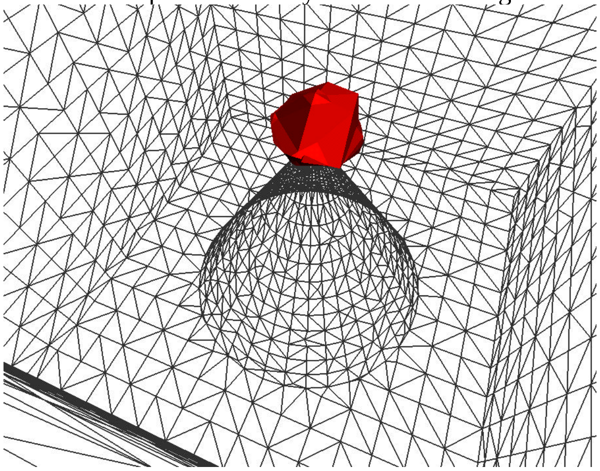

Figure 8. Surface triangles from an unadapted grid used to solve Laplace's equation on a unit cube with the bottom pierced by a sharp cone. Displayed at the tip of the cone is an isosurface of the component of the electric field aligned with the axis of the cone.

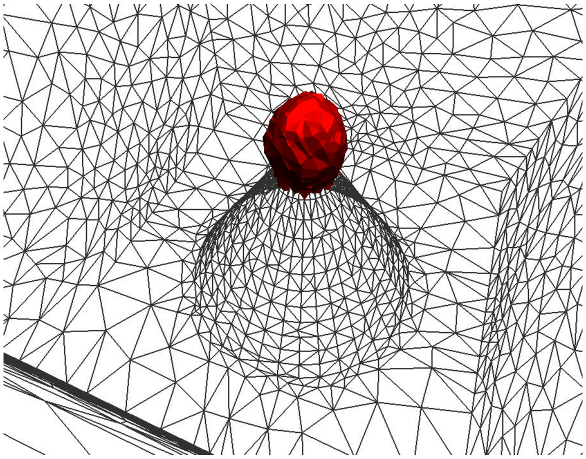

Figure 9. Surface triangles after error-based grid adaption. Note improvement in the electric field isosurface.

Thus, without increasing the number of nodes used, the solution adaption using the toolbox commands was automatically able to increase the quality of the solution in the critical area.

4: Grid Server Architecture

Because of the variety of applications that use LaGriT to provide high-quality unstructured grids, the data structures were designed as objects to be easily user extensible. We define a basic mesh object that consists of the minimum set of grid information. By either issuing a LaGriT command or by calling a subroutine, a user extends the basic mesh objects with added attributes. LaGriT maintains a table of attribute definitions for each defined mesh object. LaGriT supports standard interface file formats, such as AVS, thus providing, at a minimum, file based access to the mesh objects for application codes written in incompatible languages.

LaGriT can be operated as a stand-alone code, however it is really intended to be linked into an application driver. The application driver can issue any grid generation, maintenance, or optimization command. The previously mentioned finite volume electrostatic calculations used this approach. We extended the basic mesh object by adding attributes for the potential and for the three components of the electric field. The grid maintenance and optimization routines maintained the user defined fields, but the physics driver handled application dependent aspects such as supplying the boundary conditions to the solver. An additional benefit of this extensible design is that the extended mesh object attributes are available to the viewer. When a user adds an attribute to a mesh object, the user will specify type, size, interpolation method, persistence and target output files. In this case GMV is used as the viewer, and the field data contained in the added attributes is available for display as in Figures 8 and 9.

Additionally, users can write application specific commands that build upon the existing set of commands. We recently implemented a new grid optimization command, **massage**, whose inputs are a desired edge length and a damage tolerance which specifies the maximum acceptable amount of material interface deformation allowable. This new command was built around the existing **refine**, **merge** (de-refinement) and **recon** (reconnection) commands. Figures 10 and 11 show the grid before and after the massage. Although only the surface grid is displayed, the entire volume grid is optimized. Initially the grid had many surface nodes and few volume nodes; this 'before' state represents five grains at an intermediate time in their evolution under mean curvature[4].

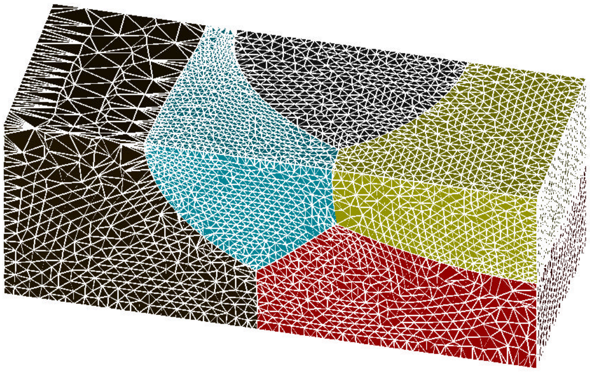

Figure 10. Evolved 5 grain microstructure showing surface grid before **massage**. Evolution has increased imbalance in grid density.

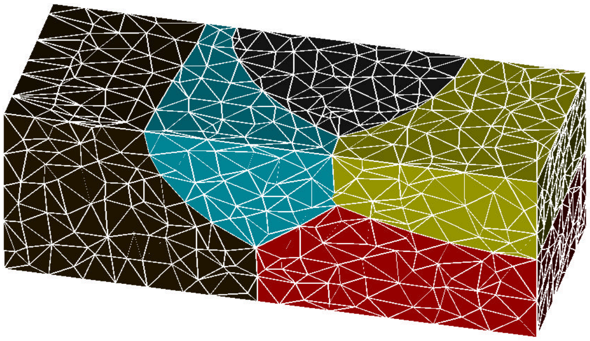

Figure 11. Five grain evolved grid after **massage**. The grid has been derefined and refined according to a desired edge length criterion while preserving material interfaces and external boundaries.

5: Conclusions:

Modeling and simulation of static and dynamic 3D processes requires generation, maintenance and optimization of high-quality 3D grids. The LaGriT toolbox produces unstructured grids appropriate for solving partial differential equations using finite volume and finite element methods. The toolbox allows separation of grid chores from the physics while providing needed access to the grid data.

References:

1. D. C. George, *LaGriT User's Manual*, maintained at http://www.t12.lanl.gov/~dcg/, previous version, *X3D User's Manual*, Los Alamos National Lab Report, LA-UR-95-3608, 1995.

2. A. Khamayseh, A. Kuprat, *Anisotropic Smoothing and Solution Adaption for Unstructured Grids,* International Journal for Numerical Methods in Engineering, Vol. 39, pp. 3163-3174, 1996.

3. A. Kuprat, *Adaptive Smoothing Techniques for 3-D Unstructured Meshes,* Los Alamos National Lab Report, LA-UR-95-1116, 1995.

4. A. Kuprat, et. al., *X3D Moving Grid Methods for Semiconductor Applications*, Los Alamos National Lab Report, LA-UR-97-2891, 1997.